Simulation techniques for parameter estimation via a stochastic ECM algorithm with applications to plant growth modeling

Samis Trevezas, Sonia Malefaki, Paul-Henry Cournède

Simulation techniques for parameter estimation via a stochastic ECM algorithm with applications to plant growth modeling

submitted

S. Trevezas[1], S. Malefaki[2], P.-H. Cournède[1]

[1]Ecole Centrale Paris, Laboratoire MAS, Châtenay Malabry, F-92295, France
INRIA Saclay, Île-de-France, EPI Digiplante, Orsay, F91893, France
Email: samis.trevezas@ecp.fr; paul-henry.cournede@ecp.fr

[2] Department of Engineering Sciences,
University of Patras, GR 26500 Rio Patras, Greece;
Email: smalefaki@upatras.gr

Abstract

Mathematical modeling of plant growth has gained increasing interest in recent years due to its potential applications. A general family of models of Carbon allocation formalized as dynamic systems serves as the basis for our study. They are known as functional-structural plant models (FSPMs, [45]). Modeling, parameterization and estimation are very challenging problems due to the complicated mechanisms involved in plant evolution. In [46] a specific type of a non-homogeneous hidden Markov model is proposed as an extension of the GreenLab FSPM ([9]) to study a certain class of plants with known organogenesis. In such a model, the maximum likelihood estimator cannot be derived explicitly. A stochastic version of an ECM (expectation conditional maximization) algorithm was adopted, where the E-step was approximated by a sequential importance sampling with resampling method (SISR-ECM approach). In this paper, a Markov Chain Monte Carlo method is proposed for the approximation of the E-step (MCMC-ECM approach). The parameter estimates obtained with MCMC-ECM are compared with those obtained with SISR-ECM from simulated and real sugar beet data. Based on this real data set competing models are compared via model selection techniques. Moreover, a data-driven automated MCMC-ECM algorithm for finding the proper sample size in each ECM step and also the proper number of ECM steps is proposed. The MCMC approach seems to be more flexible for this particular application context and can be more easily generalized to the parameter estimation of other plant models for which observations are taken under destructive measurements.

keywords: plant growth model; hidden Markov model; stochastic ECM algorithm; MCMC; Metropolis-within-Gibbs; automated Monte-Carlo EM algorithm; sequential importance sampling with resampling

1 Introduction

Mathematical modeling of plant development and growth has gained increasing interest in the last twenty years, with potential applications in agricultural sciences, plant genetics or ecology.

Due to the complicated mechanisms which guide plant's evolution, modeling, model parameterization and estimation are very challenging problems. The last decades advanced plant growth models have been proposed in the literature (see, e.g., [47]). In this paper, a certain class of plants with known organogenesis (in plants, organogenesis is the process of creation of new organs) is studied, whose growth is modeled by the so-called GreenLab functional-structural plant growth model ([9], [30]). The parameters of the model are related to plant functioning, environment, model uncertainty and observation errors. The vector of observations consists of organ masses, measured only once by censoring plant's evolution at a given observation time (destructive measurements). In [8], a first approach for parameter estimation was introduced but based on the rather restrictive assumption of an underlying deterministic model of biomass production and uncorrelated errors in the mass measurements of different organs in the plant structure. In [46] the authors proposed a more general framework for statistical analysis which can potentially be applied to a large variety of plant species by taking into account process and measurement errors. They provided a frequentist-based statistical methodology for state and parameter estimation in plants with deterministic organogenesis rules. A lot of agronomic plants can be modelled in this way, from maize ([20]) to rapeseed ([27]) or trees ([34]). This framework can also serve as the basis for statistical analysis in plant models with stochastic organogenesis as well (see also [33]). In [46], the authors proposed a non-homogeneous hidden Markov model (HMM), where the hidden states of the model correspond to the sequence of unknown biomasses (masses measured for living organisms) produced during successive growth cycles. In such a model, the maximum likelihood estimator cannot be derived explicitly and for this reason a stochastic variant of an ECM-type algorithm was adopted. The complexity of the model makes both the E-step and the M-step non-explicit. In this article, a state estimation technique based on a Markov Chain Monte Carlo (MCMC) method is proposed for the approximation of the E-step. For the M-step, we use a conditional maximization approach (see, ECM in [38]), in which the parameters of the model are separated into two groups, one for which explicit updates can be derived by fixing the parameters of the other group, and one for which updates are derived via numerical maximization. The parameter estimates obtained by the new method are compared in simulated and real data with the sequential importance sampling with resampling (SISR) method proposed in [46]. Moreover, a data-driven automated algorithm for finding the proper sample size in each ECM step and also the proper number of ECM steps is proposed. The new approach appears to be more flexible for this particular application context and can be more easily generalized to the parameter estimation of other plant models for which observations are taken under destructive measurements.

The paper is organized as follows. In Section 2, we review the basic assumptions of the GreenLab FSPM and we give a short description of the non-homogeneous hidden Markov model developed in [46]. We describe as well a new competing model which operates in the log-scale and give the framework for making maximum likelihood estimation feasible within the framework of EM-type algorithms. In Section 3, we describe the MCMC approximation to the Q-function of the E-step and compare the current approach based on MCMC with the one based on SISR. Automated Monte-Carlo EM algorithms are reviewed in Section 4, and the adaptation of the automated algorithm of [4] in our context is also given. The resulting automated MCMC-ECM is compared with the non-automatic one in synthetic examples. In Section 5, the performance of the aforementioned algorithms is tested on data from the sugar-beet plant and a model comparison is also achieved. Finally, in the last section an extended discussion is provided.

2 Description of the plant growth model

In this section we recall the basic assumptions of the GreenLab model and its formulation as an HMM given in [46]. Additionally, we propose a new candidate model and describe an appropriate version of an ECM-algorithm for maximum likelihood estimation. Starting with the initial mass of the seed q_0, plant development is considered to be the result of a cyclic procedure. The cycle duration is determined by the thermal time needed to set up new organs in the plant and is called Growth Cycle (GC). At each GC the available biomass is allocated to organs in expansion and at the same time new biomass is produced by the green (non-senescent) leaves and will be available for allocation at the next GC. The set of different classes (types) of organs of a plant are denoted by $\mathcal{O}$. In our application context with the sugar-beet plant $\mathcal{O} = \{b, p, r\}$, where b stands for blade, p for petiole and r for the root. Let us now give the assumptions concerning biomass production and allocation.

2.1 Modeling assumptions

In the sequel, we use the compact notation $x_{i:j}$ for vectors $(x_i, \ldots, x_j)$, where $i \leq j$.

Assumption 1. i) At the n-th GC, denoted by $GC(n)$, the produced biomass q_n is fully available for allocation to all expanding (preexisting + newly created) organs and it is distributed proportionally to the class-dependent empirical sink functions given by

$$s_o(i\,; p_{al}^o) = p_o\, c(a_o, b_o) \left(\frac{i+0.5}{t_o}\right)^{a_o-1} \left(1 - \frac{i+0.5}{t_o}\right)^{b_o-1}, \quad i \in \{0, 1, \ldots, t_o - 1\}, \tag{1}$$

where $p_{al}^o = (p_o, a_o, b_o) \in \mathbb{R}_+^* \times [1, +\infty)^2$ is the class specific parameter vector with $(p_o)_{o\in\mathcal{O}}$ a vector of proportionality constants representing the sink strength of each class (by convention $p_b = 1$), t_o is the expansion time period for organs belonging to the class $o \in \mathcal{O}$ and $c(a_o, b_o)$ is the normalizing constant of a discrete Beta(a_o, b_o) function, where its unnormalized generic term is given by the product of the two last factors of (1).
ii) As in [30], we suppose that expansion durations are the same for blades and petioles and T denotes their common value: $t_b = t_p = T$.

We denote by $p_{al} \triangleq (p_{al}^o)_{o\in\mathcal{O}}$ the vector of all allocation parameters and $(N_n^o)_{o\in\mathcal{O}}$ the vector of organs preformed at $GC(n)$, for all $n \in \mathbb{N}$ (determined by plant organogenesis, and deterministic in this study).

Definition 1. The total biomass demand at $GC(n)$, denoted by d_n, is the quantity expressing the sum of sink values of all expanding organs at $GC(n)$.

Since we consider that there is only one root compartment and the fact that an organ is in its i-th expansion stage if and only if (iff) it has been preformed at $GC(n-i)$ (see Assumption 1), we have that

$$d_n(p_{al}) \;=\; \sum_{o\in\mathcal{O}-\{r\}} \sum_{i=0}^{\min(n,T-1)} N_{n-i}^o\, s_o(i; p_{al}^o) + s_r(n; p_{al}^r). \tag{2}$$

Except for the initial mass of the seed q_0 subsequent biomasses $\{q_n\}_{n\geq 1}$ are the result of photosynthesis by leaf blades.

Definition 2. i) The photosynthetically active blade surface at $GC(n+1)$, denoted by s_n^{act}, is the quantity expressing the total surface area of all leaf blades that have been preformed until $GC(n)$ and will be photosynthetically active at $GC(n+1)$,
ii) the ratio (percentage) of the allocated q_l which contributes to s_n^{act} will be denoted by $\pi_{l,n}^{\text{act}}$.

Assumption 2. i) The initial mass of the seed q_0 is assumed to be fixed and known,
ii) the leaf blades have a common photosynthetically active period which equals T,
iii) the leaf blades have a common surfacic mass denoted by e_b.

Now, we describe how biomasses $\{q_n\}_{n\geq 1}$ are obtained.

Assumption 3. In the absence of modeling errors, the sequence of produced biomasses $\{q_n\}_{n\geq 1}$ is determined by the following recurrence relation known as the empirical Beer-Lambert law (see [20]):

$$q_{n+1} = F_n(q_{(n-T+1)^+:n}, u_{n+1}; p) = u_{n+1}\,\mu\, s^{pr}\left\{1 - \exp\left(-k_B \frac{s_n^{act}(q_{(n-T+1)^+:n}; p_{al})}{s^{pr}}\right)\right\}, \quad (3)$$

where $x^+ = \max(0, x)$, u_n denotes the product of the photosynthetically active radiation during $GC(n)$ modulated by a function of the soil water content, $p \triangleq (\mu, s^{pr}, k_B, p_{al})$, μ is the radiation use efficiency, s^{pr} is a characteristic surface that represents the two-dimensional projection on the ground, of space potentially occupied by the plant, k_B is the extinction coefficient in the Beer-Lambert extinction law, s_n^{act} is given by

$$s_n^{act}(q_{(n-T+1)^+:n}; p_{al}) = e_b^{-1} \sum_{l=(n-T+1)^+}^{n} \pi_{l,n}^{act}(p_{al})\, q_l, \quad (4)$$

and

$$\pi_{l,n}^{act}(p_{al}) = \frac{1}{d_l(p_{al})} \sum_{j=0}^{\min(l, l+T-n-1)} N_{l-j}^b s_b(j; p_{al}^b), \qquad (n-T+1)^+ \leq l \leq n, \quad (5)$$

where d_l is given by (2), s_b by (1) and N_n^b is the number of blades preformed at $GC(n)$.

Note that q_{n+1} also depends on p_{al}, but only through s_n^{act}, and that p could have lower dimension if some of the aforementioned parameters are fixed or calibrated in the field.

In [46] the available data Y were rearranged sequentially into sub-vectors Y_n by taking into account the preformation time (one GC before their first appearance) of all available organs except for the root mass which was excluded from the data vector. In this paper we use the same data decomposition and we also indicate a way to take into account the root mass. Each sub-vector Y_n contains the masses of the organs which are preformed at $GC(n)$. Whenever the root mass is included, it is contained in the last sub-vector. If we denote by G_n the vector-valued function that expresses the theoretical masses of all the different classes of organs which started their development at $GC(n)$, then by summing the allocated biomass at each expansion stage and Assumption 1 we obtain directly

$$G_n(q_{n:(n+T-1)}; p_{al}) = \left(\sum_{j=0}^{T-1} \frac{q_{j+n}}{d_{j+n}(p_{al})} s_o(j; p_{al}^o)\right)_{o\in O-\{r\}}. \quad (6)$$

The theoretical root mass, whenever included, is given by

$$G_r(q_{0:N}; p_{al}) = \sum_{j=0}^{N} \frac{q_j}{d_j(p_{al})} s_r(j; p_{al}^r). \tag{7}$$

The following assumptions determine the stochastic nature of the model.

Assumption 4. Let $(W_n)_{n\in\mathbb{N}}$ and $(V_n)_{n\in\mathbb{N}}$ be two mutually independent sequences of i.i.d. random variables and vectors respectively, independent of Q_0, where $W_n \sim \mathcal{N}(0, \sigma^2)$ and $V_n \sim \mathcal{N}_d(0, \Sigma)$, with Σ an unknown covariance matrix and d the cardinality of $\mathcal{O} - \{r\}$. By setting $N_n^o = 1$, $\forall o \in \{b, p\}$ two type of model equations will be assumed and compared in the sequel:

a) model $\mathcal{M}_1$: for $n \geq 0$,

$$\begin{aligned} Q_{n+1} &= F_n(Q_{(n-T+1)^+:n}; p)(1 + W_n), & (8)\\ Y_n &= G_n(Q_{n:(n+T-1)}; p_{al}) + V_n, & (9) \end{aligned}$$

b) model $\mathcal{M}_2$: for $n \geq 0$,

$$\begin{aligned} Q_{n+1} &= F_n(Q_{(n-T+1)^+:n}; p)\, e^{W_n}, & (10)\\ Y_n &= G_n(Q_{n:(n+T-1)}; p_{al}) \circ e^{V_n}, & (11) \end{aligned}$$

where F_n is given by (3), G_n is given by (6), $e^x \triangleq (e^{x_1}, \ldots, e^{x_d})$ for a d-dimensional vector $x = (x_1, \ldots, x_d)$ and $x \circ y$ is the Hadamard (entrywise) product of two vectors.

Remark 1. i) Assumption 4-a) corresponds to the model equations adopted in [46].
ii) When a data set $Y_{0:N}$ is available and the root mass is included, then the dimension of Y_N, G_N and V_N given in (9) or (11) is increased by one to incorporate the root mass given by (7) observed with error $V_{n,d+1} \sim \mathcal{N}(0, \sigma_r^2)$.

Both models given above correspond to state-space models with state sequence $\mathbf{Q}$, satisfy Assumptions 1-3, and differ in the state and observation equations given by Assumption 4-a) or 4-b).

Now, we give their equivalent formulation as hidden Markov models (HMM), see [5]. The proof is direct and will be omitted.

Proposition 1. *Under Assumptions 1-4, the bivariate stochastic process $(\mathbf{Q}, \mathbf{Y})$ defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P}_\theta)$, where $\theta = (p, \Sigma)$ or (p, σ^2, Σ) can be represented as an HMM, where*

i) the hidden sequence $\mathbf{Q}$, with values in $\mathbb{R}_+$, evolves as a time-inhomogeneous T-th order Markov chain with initial distribution $\mathbb{P}_\theta(Q_0 \in \cdot) = \delta_{q_0}(\cdot)$ (dirac at q_0), where $q_0 \in \mathbb{R}_+^*$, and transition dynamics due to Assumption 4-a) for model $\mathcal{M}_1$:

$$\mathbb{P}_\theta(Q_{n+1} \in \cdot \mid Q_{(n-T+1)^+:n}) \approx \mathcal{N}\left(F_n(Q_{(n-T+1)^+:n}; p), \sigma^2 F_n^2(Q_{(n-T+1)^+:n}; p)\right), \tag{12}$$

and due to Assumption 4-b) for model $\mathcal{M}_2$:

$$\mathbb{P}_\theta(Q_{n+1} \in \cdot \mid Q_{(n-T+1)^+:n}) = \log\mathcal{N}\big(\log F_n(Q_{(n-T+1)^+:n}; p), \sigma^2\big), \tag{13}$$

where $\log\mathcal{N}$ stands for the log-normal distribution,

ii) the observable sequence $\mathbf{Y}$*, with values in* $(\mathbb{R}_+)^d$*, conditioned on* $\mathbf{Q}$ *forms a sequence of conditionally independent random vectors and each* Y_n *given* $\mathbf{Q}$ *depends only on* $Q_{n:(n+T-1)}$ *with conditional distribution due to Assumption 4-a) for model* $\mathcal{M}_1$*:*

$$\mathbb{P}_\theta(Y_n \in \cdot \mid Q_{n:(n+T-1)}) \approx \mathcal{N}_d\left(G_n(Q_{n:(n+T-1)}; p_{al}), \Sigma\right), \tag{14}$$

and due to Assumption 4-b) for model $\mathcal{M}_2$*:*

$$\mathbb{P}_\theta(Y_n \in \cdot \mid Q_{n:(n+T-1)}) = \log \mathcal{N}_d\left(\log G_n(Q_{n:(n+T-1)}; p_{al}), \Sigma\right), \tag{15}$$

where $\log x \triangleq (\log x_1, \ldots, \log x_d)$ *for a d-dimensional vector* $x = (x_1, \ldots, x_d)$*.*

Remark 2. The model $\mathcal{M}_1$ is the one assumed in [46] and normality in (12) and (14) is only valid approximately (with small variances) since we deal with positive r.v.

2.2 Maximum likelihood estimation

The available data $y_{0:N}$ contain organ masses, measured at a given $GC(N)$ by censoring plant's evolution (destructive measurements). Based on the data set $y_{0:N}$ parameter estimation is crucial to estimate the important biophysiological parameters given by the vector p. In [8] a parameter identification method was proposed for the GreenLab model in the absence of modeling errors in biomass production ($\sigma^2 = 0$) and correlation (diagonal covariance matrix Σ) in the mass measurements. In [46] the method was extended to cover the case of a special type of modeling errors and to introduce correlation in the mass measurements. The authors make parameter estimation feasible with the help of an appropriate stochastic variant of a generalized EM-algorithm (Expectation-Maximization), see [12], [38], [23], [37]. Each iteration of an EM algorithm consists of an expectation step (E-step) and a maximization step (M-step). The E-step involves the computation of the conditional expectation of the complete data log-likelihood given the observed data under the current parameter value (called Q-function). In the M-step, the parameters are updated by maximizing the Q-function of the E-step. When the integral involved in the E-step is analytically intractable, then the Q-function, denoted here by $\mathcal{Q}(\theta; \theta')$, should be approximated. Several efforts have been made in this direction, e.g., the Stochastic EM (SEM) ([6]), the Monte Carlo EM (MCEM) ([48]), the Stochastic Approximation EM (SAEM) ([11]), as well as the Quasi-Monte Carlo EM ([22]). The common characteristic of the aforementioned variants is the approximation of the Q-function by simulating the hidden state sequence from its conditional distribution given the observed data, see [23] and [24]. In the context of hidden Markov models ([5]) the two most popular and efficient simulation mechanisms concern sequential importance sampling with resampling (SISR), see [19], [13], [5], and Markov chain Monte-Carlo, see [39], [21],[17], [16]. The resulting algorithms will be referred to as the SISR-EM and MCMC-EM algorithm. In order to perform the E-step for the HMM $\mathcal{M}_1$ the authors in [46] approximate the Q-function via a SISR estimate $\hat{\mathcal{Q}}(\theta; \theta')$. In the next section we propose an approximation of the Q-function based on MCMC. We can express this estimate in a unified way as:

$$\hat{\mathcal{Q}}(\theta; \theta') = \sum_{i=1}^{M} w_i \log p_\theta(q_{0:N}^{(i)}, y_{0:N}), \tag{16}$$

where $p_\theta(q_{0:N}, y_{0:N})$ is the density function of the complete model when the true value is θ and $\{w_i, q_{0:N}^{(i)}\}$ is a weighted M-sample (w_i, $q_{0:N}^{(i)}$ and M depend on θ') from the conditional distribution of the hidden states $q_{0:N}$ given the observed data $y_{0:N}$ when the true parameter is θ'. In the case of an MCMC estimate the weights w_i equal $1/M$.

Very often in real-life applications the M-step is analytically intractable as well. Unfortunately, any stochastic EM-type algorithm that can be designed for the HMMs $\mathcal{M}_1$ and $\mathcal{M}_2$ given by Proposition 1 leads to a non-explicit M-step as well. For this reason, a numerical maximization procedure of quasi-Newton type could be implemented at each iteration of a stochastic EM algorithm (see [46]) in the same way as it is implemented in a deterministic EM algorithm (see [29]). Nevertheless, it is certainly desirable, whenever possible, for time and accuracy reasons to reduce the number of parameters to be updated via numerical maximization. A way to overcome a complicated M-step was proposed in [38] with the so-called ECM (Expectation Conditional Maximization) algorithm, where the authors separated the intractable M-step into smaller tractable conditional M-steps and updated in a cyclic fashion the parameters of the model. In order to perform the M-step for the HMM $\mathcal{M}_1$ the authors in [46] combined conditional and numerical maximization. First, they updated explicitly in a CM (conditional maximization) step the parameters which have explicit updates given fixed values of the rest and then updated the rest by the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton algorithm. This approach is also adopted here for both models $\mathcal{M}_1$ and $\mathcal{M}_2$. Let $\hat{\mathcal{Q}}(\theta;\theta^{(t)})$ denote the approximation of $Q(\theta;\theta^{(t)})$ given by (16) in the t-th EM-iteration ($\theta' = \theta^{(t)}$) and (θ_1, θ_2) be a decomposition of θ in two subvectors, where θ_1 can be explicitly updated given θ_2. The maximization of $\hat{\mathcal{Q}}(\theta;\theta^{(t)})$ with respect to $\theta = (\theta_1, \theta_2)$ is described by the following two steps:

$$\begin{aligned} \theta_1^{(t+1)} &= \arg\max_{\theta_1} \hat{\mathcal{Q}}(\theta_1, \theta_2^{(t)}; \theta_1^{(t)}, \theta_2^{(t)}), \\ \theta_2^{(t+1)} &= \mathrm{BFGS}\max_{\theta_2} \hat{\mathcal{Q}}(\theta_1^{(t+1)}, \theta_2; \theta_1^{(t)}, \theta_2^{(t)}), \end{aligned} \tag{17}$$

where the notation BFGS max corresponds to the solution of the maximization problem with the BFGS quasi-Newton algorithm. The explicit step (17) corresponding to model $\mathcal{M}_1$ can be found in [46]. The solution to (17) for the model $\mathcal{M}_2$ is given below. The proof is deferred to the Appendix.

Proposition 2. *Let $\theta = (\theta_1, \theta_2)$, where $\theta_1 = (\mu, \Sigma)$ and θ_2 contains all parameters of p except for μ. The update equations for θ_1 are given as follows:*

$$\widehat{\mu}_N(\theta_2;\theta') = \mu' \exp\left\{ N^{-1} \sum_{n=1}^{N} \mathbb{E}_{\theta'} \left[\log Q_n - \log F_{n-1}(\theta_2) \mid y_{0:N} \right] \right\}, \tag{18}$$

$$\widehat{\Sigma}_N(\theta_2;\theta') = (N+1)^{-1} \sum_{n=0}^{N} \mathbb{E}_{\theta'} \left[\big(\log y_n - \log G_n(\theta_2) \big) \big(\log y_n - \log G_n(\theta_2) \big)^\top \mid y_{0:N} \right]. \tag{19}$$

If σ^2 is estimated as well, then its update equation is given by:

$$\widehat{\sigma_N^2}(\theta_2;\theta') = N^{-1} \sum_{n=1}^{N} \mathbb{E}_{\theta'} \left[\Big(\log Q_n - \log F_{n-1}(\theta_2) + \log \mu' \Big)^2 \mid y_{0:N} \right] - \big(\log \widehat{\mu}_N(\theta_2;\theta') \big)^2. \tag{20}$$

3 MCMC approximation of the Q-function

In this section we propose a suitable approximation of the Q-function by using an MCMC algorithm (the weights in (16) are equal) and we compare this approach with the one based on SISR developed in [46].

3.1 E-step via Markov Chain Monte Carlo

At each iteration of the EM-algorithm, the basic problem is to sample in the most effective way from $p_{\theta'}(q_{1:N} \mid q_0, y_{0:N})$, where θ' is the current estimation of the model parameters. For the rest, we alleviate the index θ' since we focus on the general sampling problem (θ' is known and fixed at each iteration). Thus, conditionally on $Q_0 = q_0$ and $Y_{0:N} = y_{0:N}$, the hidden states are sampled from:

$$Q_{1:N} \sim p(q_{1:N}|q_0, y_{0:N}) \propto p(q_{1:N}, y_{0:N}|q_0). \tag{21}$$

One of the most important MCMC algorithms for sampling from a multidimensional distribution (such as (21)) is Gibbs sampler ([17], [16]). Gibbs Sampler uses only the full conditional distributions in order to sample a Markov chain with stationary distribution corresponding to the multidimensional target one. For brevity, when we explicit densities in the sequel, we refer only to model $\mathcal{M}_1$ since the general approach that we consider here is entirely the same for both models $\mathcal{M}_1$ and $\mathcal{M}_2$. The full conditional distribution of Q_n given all the other variables, denoted by $\pi_n(q_n|q_{0:n-1}, q_{n+1:N}) \triangleq p(q_n|q_{0:n-1}, q_{n+1:N}, y_{0:N})$ corresponding to model $\mathcal{M}_1$, by (12) and (14) can be written in the form:

$$\begin{aligned}\pi_n(q_n|q_{0:n-1}, q_{n+1:N}) \quad \propto \quad & \prod_{i=n+1}^{(n+T)\wedge N} \left(1 - \exp\left\{-\delta\, s_{i-1}^{act}(q_n)\right\}\right)^{-1} \\ & \times \exp\left\{-\frac{1}{2\sigma^2}\left(\left(\frac{q_n}{F_{n-1}(q_n)} - 1\right)^2 + \sum_{i=n+1}^{(n+T)\wedge N}\left(\frac{q_i}{F_{i-1}(q_n)} - 1\right)^2\right)\right\} \\ & \times \exp\left\{-\frac{1}{2}\sum_{i=(n-T+1)^+}^{n} (y_i - G_i(q_n))^\top \Sigma^{-1} (y_i - G_i(q_n))\right\}, \end{aligned} \tag{22}$$

where $\delta = k_B/s^{pr} > 0$, see (3), and all the other quantities that appear in this expression are explained in Section 2 and are expressed here only as functions of q_n.

Clearly, direct simulation from (22) is impossible. For this reason alternative sampling techniques are required, such as hybrid Gibbs Sampler. Hybrid Gibbs Sampler is a Gibbs Sampler where at least one of the simulations of the full conditional distributions is replaced by a Metropolis–Hastings (MH) step ([39], [21]). Let π_n, $n = 1, \ldots, N$, be the densities of the unnormalized full conditional distributions given by (22) and $f_n(z_n|q_{1:n-1}, q_{n+1:N})$, $i = 1, \ldots, N$ be the densities of the proposal distributions.

Let also

$$\alpha(q_n^{t-1}, z_n) = \min\left\{1, \frac{\pi_n(z_n|q_{1:n-1}^t, q_{n+1:N}^{t-1})}{\pi_n(q_n^{t-1}|q_{1:n-1}^t, q_{n+1:N}^{t-1}} \cdot \frac{f_n(q_n^{t-1}|q_{1:n-1}^t, z_n, q_{n+1:N}^{t-1})}{f_n(z_n|q_{1:n-1}^t, q_n^{t-1}, q_{n+1:N}^{t-1})}\right\}$$

denote the acceptance probability of the MH-step. The hybrid Gibbs Sampler can be described as follows:

Initialize $q_{1:N}^0$
`For` $t = 1$ `to` M
  `For` $n = 1$ `to` N
    Draw $z_n \sim f_n(q_n|q_{1:n-1}^t, q_{n+1:N}^{t-1})$
    Set $q_n^t = z_n$ with probability $\alpha(q_n^{t-1}, z_n)$

otherwise set $q_n^t = q_n^{t-1}$

End

End

The choice of the proposal distribution is not of great importance for the convergence of the Markov chain to the target distribution. The proposal distribution can be chosen arbitrarily with the limitation to satisfy the conditions that ensure the convergence to the target distribution ([40], [43]). On the other hand, the proposal distribution affects the convergence rate of the Markov chain to the target distribution. The more the proposal is closer to the target, the faster the desirable convergence is achieved. Moreover, it must be easy and fast to sample from. In this paper, we used as a proposal distribution for the hidden states the one resulting from the prior transition kernel of the hidden chain under the current parameters' values given by (12) for model $\mathcal{M}_1$ and by (13) for model $\mathcal{M}_2$. We also tried a Random-Walk Metropolis-Hastings with different variances and the results that we obtained were worse.

3.2 Comparing MCMC and SISR using simulated datasets

In order to evaluate the effect of the MCMC approximation of the Q-function in parameter estimation and to compare this approach with the one proposed by [46] using SISR, we performed several tests with simulated data. Here, we present a comparison for the model $\mathcal{M}_1$. We generated one data vector $y_{0:N}$ with $N = 50$ for several values of σ and we present the cases where $\sigma \in \{0.02, 0.1\}$. The parameters' values that we used to simulate the data are presented in Table 1.

Table 1: Parameters' values used to generate the data (for $\sigma \in \{0.02, 0.1\}$), where σ_b, σ_p and ρ are the standard deviations and the correlation coefficient of the measurement error model. The explanation of the other parameters is given in Section 2. The parameters that should be estimated are given in the first column.

param.	unknown	param.	known	param.	known
a_b	3	q_0	0.003	e_b	0.0083
a_p	3	a_r	5.5	b_r	2
p_p	0.8165	p_r	400	b_b	2
μ^{-1}	100	k_B	0.7	b_p	2
σ_b	0.05	t_r	100		
σ_p	0.05	T	10		
ρ	0.8	s^{pr}	500		

As a stopping criterion for the EM algorithm we used a predefined number of EM steps (100 EM steps). For each independent run of the algorithm, the sample size was increased piecewise linearly (with increasing slope) for the first 50 iterations (starting from 250, then increased by 10 for the first 25 iterations and by 20 for the subsequent 25 iterations), and for the last 50 iterations we used a quadratic increase until we reached 10.000 trajectories. The burn-in period for the MCMC was fixed at 500 iterations. For a similar type of simulation schedule and some discussion on some alternatives see [5]. Our final estimates for both algorithms were based on means from 50 independent runs. We also tested the effect of the averaging technique developed by [15] (see also [5], p. 407). The authors proposed to smooth the estimates from each

independent run by averaging after a burn-in period all subsequent EM-updates with weights proportional to the Monte-Carlo sample size used in the corresponding EM iterations. This technique is typically used when the simulation noise at convergence is still significant. We present 3 different scenarios: no-averaging and averaging from the last 25 or 50 iterations. The means and the standard deviations of the final estimates based on the 50 independent runs are presented in Tables 2-3 for both values of $\sigma \in \{0.02, 0.1\}$.

Table 2: Parameter estimation results for the synthetic example when $\sigma = 0.02$. Means and standard deviations of the estimates based on 50 independent runs with SISR-ECM and MCMC-ECM for three different averaging strategies.

	No averaging		Averaging 25		Averaging 50	
param.	SISR	MCMC	SISR	MCMC	SISR	MCMC
a_b	2.9703	2.9705	2.9706	2.9706	2.9706	2.9706
	$(9.48 \cdot 10^{-4})$	$(5.53 \cdot 10^{-4})$	$(7.48 \cdot 10^{-4})$	$(4.82 \cdot 10^{-4})$	$(7.98 \cdot 10^{-4})$	$(5.08 \cdot 10^{-4})$
a_p	2.9714	2.9716	2.9716	2.9716	2.9716	2.9716
	$(8.45 \cdot 10^{-4})$	$(5.08 \cdot 10^{-4})$	$(6.59 \cdot 10^{-4})$	$(4.07 \cdot 10^{-4})$	$(7.01 \cdot 10^{-4})$	$(4.32 \cdot 10^{-4})$
p_p	0.8153	0.8153	0.8153	0.8153	0.8153	0.8153
	$(0.10 \cdot 10^{-4})$	$(0.04 \cdot 10^{-4})$	$(0.02 \cdot 10^{-4})$	$(0.01 \cdot 10^{-4})$	$(0.02 \cdot 10^{-4})$	$(0.01 \cdot 10^{-4})$
μ^{-1}	100.2885	100.2870	100.2875	100.2875	100.2876	100.2876
	$(5.53 \cdot 10^{-3})$	$(5.23 \cdot 10^{-3})$	$(3.58 \cdot 10^{-3})$	$(2.47 \cdot 10^{-3})$	$(3.98 \cdot 10^{-3})$	$(2.64 \cdot 10^{-3})$
σ_b	0.0477	0.0477	0.0477	0.0477	0.0477	0.0477
	$(0.47 \cdot 10^{-4})$	$(0.26 \cdot 10^{-4})$	$(0.14 \cdot 10^{-4})$	$(0.09 \cdot 10^{-4})$	$(0.14 \cdot 10^{-4})$	$(0.09 \cdot 10^{-4})$
σ_p	0.0504	0.0504	0.0504	0.0504	0.0504	0.0504
	$(0.45 \cdot 10^{-4})$	$(0.28 \cdot 10^{-4})$	$(0.14 \cdot 10^{-4})$	$(0.08 \cdot 10^{-4})$	$(0.14 \cdot 10^{-4})$	$(0.08 \cdot 10^{-4})$
ρ	0.8282	0.8282	0.8282	0.8282	0.8282	0.8282
	$(3.69 \cdot 10^{-4})$	$(2.35 \cdot 10^{-4})$	$(1.08 \cdot 10^{-4})$	$(0.55 \cdot 10^{-4})$	$(1.05 \cdot 10^{-4})$	$(0.57 \cdot 10^{-4})$

Table 3: Parameter estimation results for the synthetic example when $\sigma = 0.1$. Means and standard deviations of the estimates based on 50 independent runs with SISR-ECM and MCMC-ECM for three different averaging strategies.

	No averaging		Averaging 25		Averaging 50	
param.	SISR	MCMC	SISR	MCMC	SISR	MCMC
a_b	2.8719	2.8767	2.8843	2.8871	2.8806	2.8836
	$(1.59 \cdot 10^{-2})$	$(0.34 \cdot 10^{-2})$	$(1.77 \cdot 10^{-2})$	$(0.41 \cdot 10^{-2})$	$(1.78 \cdot 10^{-2})$	$(0.41 \cdot 10^{-2})$
a_p	2.8756	2.8803	2.8879	2.8906	2.8842	2.8871
	$(1.58 \cdot 10^{-2})$	$(0.35 \cdot 10^{-2})$	$(1.76 \cdot 10^{-2})$	$(0.41 \cdot 10^{-2})$	$(1.77 \cdot 10^{-2})$	$(0.40 \cdot 10^{-2})$
p_p	0.8153	0.8153	0.8153	0.8153	0.8153	0.8153
	$(0.33 \cdot 10^{-4})$	$(0.06 \cdot 10^{-4})$	$(0.06 \cdot 10^{-4})$	$(0.01 \cdot 10^{-4})$	$(0.07 \cdot 10^{-4})$	$(0.01 \cdot 10^{-4})$
μ^{-1}	100.8473	100.8097	100.7520	100.7325	100.7788	100.7577
	(0.1172)	(0.0339)	(0.1270)	(0.0299)	(0.1278)	(0.0293)
σ_b	0.0504	0.0505	0.0505	0.0505	0.0505	0.0505
	$(2.41 \cdot 10^{-4})$	$(1.78 \cdot 10^{-4})$	$(0.65 \cdot 10^{-4})$	$(0.56 \cdot 10^{-4})$	$(0.67 \cdot 10^{-4})$	$(0.58 \cdot 10^{-4})$
σ_p	0.0535	0.0536	0.0536	0.0537	0.0536	0.0536
	$(2.28 \cdot 10^{-4})$	$(2.29 \cdot 10^{-4})$	$(0.93 \cdot 10^{-4})$	$(0.74 \cdot 10^{-4})$	$(0.91 \cdot 10^{-4})$	$(0.79 \cdot 10^{-4})$
ρ	0.8527	0.8533	0.8534	0.8535	0.8533	0.8534
	$(1.62 \cdot 10^{-3})$	$(1.67 \cdot 10^{-3})$	$(7.06 \cdot 10^{-4})$	$(5.34 \cdot 10^{-4})$	$(6.93 \cdot 10^{-4})$	$(5.75 \cdot 10^{-4})$

In Tables 2-3, we remark that SISR-ECM and MCMC-ECM give similar means for both values of σ and the means are closer when σ is smaller. If we use averaging for the estimation, then the estimates are even closer. The resulting estimates from the MCMC-ECM algorithm

have smaller variance than the ones from the SIS-ECM algorithm for both values of σ (except from σ_p and ρ when $\sigma = 0.1$ and no averaging is used). We also noticed with some supplementary tests that as σ increases the superiority of MCMC-ECM is clearer, that is, as compared to the SISR-ECM, it gives much more concentrated estimates of the structural parameters (the first four) for independent runs of the algorithms. Notice also that the mean estimates that we obtain for the structural parameters with both algorithms are closer to the true ones (see also Table 1) when $\sigma = 0.02$, and this is very natural since as σ increases (directly related to the model uncertainty), the uncertainty for the values of the structural parameters becomes larger. As far as the averaging is concerned, we mention that it acts in a different way for the different values of σ. When $\sigma = 0.02$, averaging improves the estimators of both algorithms with respect to the standard error. It is clear that the structural parameters benefit less from the averaging than the parameters of the measurement error. In the case where $\sigma = 0.1$, averaging decreases the variability of the latter parameters, but not of all the structural parameters. Indeed, the variability in the most sensitive parameter p_p is significantly decreased but not in the other three. On the other hand, notice that the mean estimates of all structural parameters are closer to the true values when averaging is performed. This shows that averaging has generally a positive effect. Finally, if we increase the size of the averaging too much (from 25 to 50), then the improvement decreases. This is natural since averaging should be used near the convergence region and not too early.

The above conclusions are true for the given set of parameters. The two methods have been tested in several sets of parameters, in all of them, both methods returned similar mean estimates for 50 independent runs of the algorithms, which are reasonably close to the true ones. Nevertheless, their standard errors are dependent on the value of σ and on the algorithm employed. In the examples that we run, the MCMC-ECM gives smaller standard errors than the SISR-ECM except for very small σ.

Another advantage of the MCMC approach concerns the number of data taken into account for the estimation. For a large value of T, the SISR-ECM can generally take into account only some (and not all) of the organs that had not reached their full expansion stage when the plant was cut (the immature members). The reason behind this is that the underlying hidden Markov process is T-dependent and consequently the last weights associated with the particles in the sequential implementation of the SISR could degenerate before taking into account all the data. We refer to [46] for further details of this implementation. In equation (4.5) of the above reference the following result holds for the final weights of the improved filter:

$$w_{N-T+1}^{(i)} = w_{N-T}^{(i)}\, p_\theta(y_{N-T+1}|q_{N-T:N-1}^{(i)}) \prod_{n=N-T+2}^{N} p_\theta(y_n|q_{n:N-1}^{(i)}, \tilde{q}_N^{(i)}),$$

where $\{w_{N-T}^{(i)}, q_{N-T:N-1}^{(i)})\}_{i=1}^{M}$ stands for the available weighted sample one iteration before the last update, and $\tilde{q}_N^{(i)}$ are the final proposed particle positions. It is clear that since the last product has $T-1$ factors, a practical implementation of this filter needs to stop the algorithm when the effective sample size (ESS) will be lower than a threshold for the first time (see [46] for the explanation of the ESS). This is the reason why some data may be lost and this could be a serious problem for large values of T. In the case that MCMC is used this problem does not exist. In this example we excluded from the data vector all the immature members in order to compare both algorithms on the same data.

4 Ascent–based MCMC-ECM algorithm

In the previous section we did not emphasize on the specification of the Monte Carlo sample size in each ECM step and/or the number of the ECM steps. It is known that, if the MC sample size remains constant at each EM iteration, the MCEM algorithm will not converge due to a persistent Monte Carlo error, unless this size is unnecessarily too large. Moreover, it is inefficient to start with a large sample size since the initial guess of the MLE may be far from the true value ([48]). Many authors use a deterministic increase in the MC sample size (independently of the data) and stop the algorithm after a predefined number of EM steps ([35], [36], [7]). Nevertheless, these are not the most effective ways to tackle these problems.

4.1 Data Driven Automated stochastic EM algorithms

The last decades data–driven strategies have been proposed in the literature to control the Monte Carlo sample size and the number of the EM steps. In [2] proposed an automated procedure to determine at each step if the sample size should be increased or not. This procedure concerns those Monte Carlo EM algorithms for which the random sample in the E–step is simulated either by exact draws from the corresponding conditional distribution or by importance sampling from a proposal distribution close enough to the exact one. Based on the random sample of each step t, an asymptotic confidence interval about the current estimate of the parameter $\theta^{(t)}$ is constructed. If the past value $\theta^{(t-1)}$ lies in it, then the EM step is said to be swamped by the Monte Carlo error and the sample size is increased. The size of the additional sample is arbitrary (e.g. $m_t \to m_t + m_t/c,\ c = 2, 3, \ldots$). Moreover, in [2] proposed to stop the MCEM algorithm when a stopping rule is satisfied for three consecutive iterations. The most commonly used stopping criterion is a sufficiently small relative change in the parameters' values.

The automated Monte Carlo EM algorithm of [2] was generalized from random to dependent samples by [31]. One basic difficulty which arises with dependent samples is how to determine the aforementioned confidence interval. In this direction, the authors in [31] evoke a central limit theorem (see Theorem 1, [31]) on the basis of the subsampling scheme of [44]. In particular, the Monte Carlo sample size is increased, if at least one of the estimated partial derivatives of the Q-function with respect to $\theta^{(t-1)}$, computed on the basis of the subsample, lies in the appropriately designed confidence interval.

Following the steps of [2] and [31], the authors in [32] proposed an alternative automated MCMC-EM algorithm. The method of increasing the sample size is based as well on the construction of an appropriate confidence interval. The main innovation of this paper is that the authors give a specific formula for quantifying the increase in the MC sample size. In this approach, the EM procedure should be applied two times at each iteration, one for the complete sample and one for the subsample. This is not an issue when the overall implementation of the EM algorithm is not time consuming, but if, for example, a numerical maximization is needed for the M–step, this method could be computationally expensive.

In the rest of this subsection we present the data–driven automated MCEM algorithm proposed by [4] which is computationally cheap and can be easily adapted in our case where numerical maximization is involved as well. Now, we give a short description of the basic ideas of the algorithm. Let

$$\Delta Q = \mathcal{Q}(\theta^{(t)}; \theta^{(t-1)}) - \mathcal{Q}(\theta^{(t-1)}; \theta^{(t-1)}), \tag{23}$$

$$\Delta \hat{Q} = \hat{\mathcal{Q}}(\theta^{(t)}; \theta^{(t-1)}) - \hat{\mathcal{Q}}(\theta^{(t-1)}; \theta^{(t-1)}), \tag{24}$$

where $\mathcal{Q}$ corresponds to the true Q–function of the model and $\hat{\mathcal{Q}}$ to the proxy given by (16), where the approximation is based on the m_t-sample generated at the t-th iteration of the EM. The most important feature of this algorithm is that it is an Ascent-based Monte-Carlo EM algorithm, since the basic focus is to recover with high probability the ascent property of the EM. This means that the MC sample size should be chosen throughout iterations in such a way that $\Delta Q > 0$ with high probability. The authors claim that $\Delta\hat{Q}$ is a strongly consistent estimator of ΔQ and by evoking the appropriate central limit theorem the following asymptotic result holds true

$$\sqrt{m_t}\,(\Delta\hat{Q} - \Delta Q) \xrightarrow[m_t\to\infty]{d} \mathcal{N}(0, \sigma_Q^2), \tag{25}$$

where the regularity conditions and the asymptotic variance σ_Q^2 depend on the sampling mechanism employed. A sketch of the proof is given in the case that simulations result from i.i.d. draws and a remark is made that if an MCMC algorithm is employed, then (25) holds true under stringent regularity conditions. With the help of (25) and a consistent estimator $\hat{\sigma}_Q^2$ of σ_Q^2 the following asymptotic lower bound (ALB) with confidence level $1-\alpha$ can be given for ΔQ:

$$ALB = \Delta\hat{Q} - \frac{\hat{\sigma}_Q}{\sqrt{m_t}}\, z_\alpha, \tag{26}$$

where z_α is the upper α-quantile of the standard normal distribution. In the same way, an asymptotic upper bound (AUB) with confidence level $1-\gamma$ can also be obtained for ΔQ:

$$AUB = \Delta\hat{Q} + \frac{\hat{\sigma}_Q}{\sqrt{m_t}}\, z_\gamma. \tag{27}$$

The authors use (26) to decide if the current update based on the m_t-sample will be accepted or not. In particular:

- if $ALB > 0$, then with high probability $\Delta Q > 0$ and $\theta^{(t)}$ is accepted as the new update,
- if $ALB \leq 0$, then $\hat{Q}$ is said to be swamped with MC error and a new sample is appended to the existing one to obtain a new parameter estimate. A geometric increase is recommended (e.g., $m_t \to m_t + m_t/k,\ k = 2, 3, \ldots$). The process is repeated until $ALB > 0$ for the first time.

After acceptance of $\theta^{(t)}$, the MC sample size for the next MCEM step is determined by using the approximation

$$\Delta\hat{Q}_{t+1} \sim \mathcal{N}\left(\Delta\hat{Q}_t, \frac{\hat{\sigma}_Q^2}{m_{t+1}}\right), \tag{28}$$

where $\hat{Q}_t$ is given by (24) and $\hat{Q}_{t+1}$ corresponds to the same quantity by letting $t \to t+1$. Indeed, the size m_{t+1} is chosen in such a way so as to prespecify the probability to reject the estimate $\theta^{(t+1)}$ ($ALB < 0$), when $\Delta Q > 0$ (type-II error). If we set to β this probability and add the logical requirement $m_t \leq m_{t+1}$, then it can be easily shown by (28) that

$$m_{t+1} = \max\{m_t, \hat{\sigma}_Q^2 (z_\alpha + z_\beta)^2 / (\Delta\hat{Q}_t)^2\}, \tag{29}$$

where m_t corresponds to the initial MC sample size of iteration t (before any eventual augmentation) and z_β to the upper β-quantile of the standard normal distribution. The last requirement is the stopping criterion. The MCEM algorithm stops if the upper bound $AUB < \delta$, where AUB is given in (27) and δ is a predefined small constant. If this criterion is satisfied, then

the change in the Q-function is acceptably small. The adaptation of this approach in the case of the MCMC-ECM that we propose in this paper is straightforward as long as a method for estimating the variance σ_Q^2 is available.

There are several methods for estimating σ_Q^2 (see, e.g., [18]). One of the most well-known relies on the spectral estimator which involves the estimation of autocorrelations weighted by a prespecified function. We suggest [42] for a presentation of different choices of weight functions. One other popular method is batch means (BM) ([3]) which is based on the division of the MC sample into a predefined number of batches of equal size. The batch means are treated as independent which is only approximately true if the length of each batch is much longer than the characteristic mixing time of the chain. If the batch size is allowed to increase with respect to the sample size m, then this method is referred to as CBM. Usually, the batch size is set equal to $\lfloor m^l \rfloor$, where $l = 1/2$ or $l = 1/3$. An alternative method for variance estimation is based on regenerative simulation (RS) (see, [41]), where random times at which the Markov chain probabilistically restarts itself are identified. In fact, the CBM can be viewed as an ad hoc version of the RS method (see, [26]). Both methods split the sample into pieces with the difference that the RS method guarantees that the pieces are truly independent. Nevertheless, the conditions of RS are hard to verify. The different variance estimation methods are compared in several papers (see, [26], [25] and [14]). In [26], the authors concluded that CBM and RS give similar results. Despite the theoretical advantages of RS and of the spectral estimator we adopt the CBM method in the proposed algorithm which is significantly simpler and quicker in practice.

4.2 The automated MCMC-ECM algorithm in simulated datasets

In order to evaluate the performance of the automated MCMC-ECM, we performed the same synthetic tests as the ones that we presented in Subsection 3.2.

The augmentation rule for the Monte-Carlo sample size is given by (29). As a stopping criterion we used $AUB < 10^{-3}$, see (27). The initial sample size was fixed at 250 and the burn-in period at 500. Our final estimates for all the tests were based on means from 50 independent runs. In Tables 4 and 6 (when $\sigma = 0.1$ and $\sigma = 0.02$ respectively), the parameters' estimates and the corresponding standard errors are presented for different combinations of the asymptotic levels α, β and γ, see (26) and (27). For each such combination, we compared two different rates of the geometric increase in the sample size ($m_t \to m_t + m_t/k$, for $k = 2, 3$) when $ALB \leq 0$, see (26). In Table 4 we present the results for both rates given some selected asymptotic levels for the case that $\sigma = 0.1$. We remark in Table 5 that the mean total and final sample sizes are increased when we choose $k = 2$ ($+m_t/2$) instead of $k = 3$ ($+m_t/3$). Nevertheless, this does not reflect an improvement in parameter estimation or a decrease in their standard deviations (see Table 4). For this reason all the tests in the sequel are performed with $k = 3$. Moreover, in Tables 5 and 7, the corresponding descriptive statistics for the total sample size (TSS), the final sample size (FSS) and the number of ECM iterations (Iter) until convergence are given. Note that since it is an automated algorithm the final sample size and the number of iterations until convergence will differ among independent realizations. For this reason we also present in Table 6 the effect of weighting the estimates from independent runs with weights proportional to their final sample size.

For all the tested values of α, β and γ, the best results with respect to the standard errors were given in the majority of the cases for the level 0.1 and then for $0.1 - 0.25 - 0.1$ as expected (with some exceptions). This is better reflected in the parameters of the measurement error.

Table 4: Parameter estimation results with the automated MCMC-ECM algorithm for the synthetic example when $\sigma = 0.1$. Means and standard deviations of the estimates based on 50 independent runs. The results are obtained for different values of the asymptotic levels α, β and γ, see relations (26) and (27), and for two different rates of the geometric increase in the sample size ($m_t \to m_t + m_t/k$, for $k = 2, 3$) when $ALB \leq 0$, where ALB is given by (26)

α, β, γ	0.25		0.20		0.15		0.1–0.25–0.1	0.1
increase	$(+m_t/3)$	$(+m_t/2)$	$(+m_t/3)$	$(+m_t/2)$	$(+m_t/3)$	$(+m_t/2)$	$(+m_t/3)$	$(+m_t/3)$
a_b	2.8961	2.8952	2.9052	2.9028	2.9031	2.9046	2.9004	2.9057
	$(7.73 \cdot 10^{-3})$	$(9.17 \cdot 10^{-3})$	$(8.15 \cdot 10^{-3})$	$(6.45 \cdot 10^{-3})$	$(6.39 \cdot 10^{-3})$	$(7.87 \cdot 10^{-3})$	$(5.74 \cdot 10^{-3})$	$(7.31 \cdot 10^{-3})$
a_p	2.8996	2.8987	2.9087	2.9062	2.9065	2.9080	2.9039	2.9092
	$(7.67 \cdot 10^{-3})$	$(9.13 \cdot 10^{-3})$	$(8.13 \cdot 10^{-3})$	$(6.41 \cdot 10^{-3})$	$(6.34 \cdot 10^{-3})$	$(7.87 \cdot 10^{-3})$	$(5.71 \cdot 10^{-3})$	$(7.28 \cdot 10^{-3})$
P_p	0.8153	0.8153	0.8153	0.8253	0.8153	0.8153	0.8153	0.8153
	$(0.05 \cdot 10^{-4})$	$(0.06 \cdot 10^{-4})$	$(0.05 \cdot 10^{-4})$	$(0.05 \cdot 10^{-4})$	$(0.05 \cdot 10^{-4})$	$(0.05 \cdot 10^{-4})$	$(0.06 \cdot 10^{-4})$	$(0.05 \cdot 10^{-4})$
μ^{-1}	100.6717	100.6778	100.6053	100.6224	100.6205	100.6124	100.6421	100.6004
	(0.0583)	(0.0695)	(0.0432)	(0.0554)	(0.0598)	(0.04848)	(0.0494)	(0.0559)
σ_b	0.0505	0.0505	0.0505	0.0505	0.0505	0.0505	0.0506	0.0506
	$(1.48 \cdot 10^{-4})$	$(1.41 \cdot 10^{-4})$	$(1.35 \cdot 10^{-4})$	$(1.20 \cdot 10^{-4})$	$(0.82 \cdot 10^{-4})$	$(0.86 \cdot 10^{-4})$	$(0.74 \cdot 10^{-4})$	$(0.64 \cdot 10^{-4})$
σ_p	0.0537	0.0537	0.0537	0.0537	0.0537	0.0537	0.0537	0.0537
	$(2.02 \cdot 10^{-4})$	$(1.89 \cdot 10^{-4})$	$(1.79 \cdot 10^{-4})$	$(1.46 \cdot 10^{-4})$	$(1.08 \cdot 10^{-4})$	$(1.14 \cdot 10^{-4})$	$(0.86 \cdot 10^{-4})$	$(0.71 \cdot 10^{-4})$
ρ	0.8536	0.8537	0.8538	0.8536	0.8538	0.8539	0.8540	0.8540
	$(1.47 \cdot 10^{-3})$	$(1.36 \cdot 10^{-3})$	$(1.29 \cdot 10^{-3})$	$(1.05 \cdot 10^{-3})$	$(0.80 \cdot 10^{-3})$	$(0.82 \cdot 10^{-3})$	$(0.62 \cdot 10^{-3})$	$(0.52 \cdot 10^{-3})$

Table 5: Descriptive statistics for the total sample size (TSS), the final sample size (FSS) and the number of iterations until convergence (Iter) corresponding to the tests given in Table 4

α, β, γ	0.25		0.20		0.15		0.1–0.25–0.1	0.1
increase	$(+m_t/3)$	$(+m_t/2)$	$(+m_t/3)$	$(+m_t/2)$	$(+m_t/3)$	$(+m_t/2)$	$(+m_t/3)$	$(+m_t/3)$
Min TSS	49247	13670	77627	5811	120139	150406	90505	137264
Mean TSS	250800	287785	340029	374602	534498	565618	593939	794573
Max TSS	637305	637305	1042743	916609	1017042	1186568	1401011	1936174
Min FSS	3405	2031	4693	3379	7445	8813	8508	13309
Mean FSS	16495	19665	23421	27398	44455	49981	48803	65199
Max FSS	56985	55005	56678	67744	153184	111474	164524	169807
Min Iter	46	30	35	43	41	39	45	28
Max Iter	86	94	82	79	75	74	74	66

However, if we run the algorithm by setting the levels at 0.1, then a great computational cost is involved (see Tables 5 and 7) which is not compensated for the gain in precision. For this reason it could be wiser to decrease the values of the levels to have a rapid algorithm with an acceptable precision. Furthermore, the weighted averages (see Table 6) with respect to the final sample size generally decreased the standard deviations independently of the values of the asymptotic levels.

It is noteworthy that the automated algorithm gives mean estimates which are closer to the real ones than the original MCMC-ECM algorithm. On the other hand, even the "best" automated algorithm gives more variable estimates than the non-automated one with independent runs of the algorithm. This could be expected from the variability in the final sample size and the number of iterations until convergence of the automated algorithm. The main point here is that the resulting estimators are of acceptable accuracy in less ECM steps and thus in less CPU time if the asymptotic levels are not set too low. This is very important for a routine use of the

Table 6: Parameter estimation results with the automated MCMC-ECM algorithm for the synthetic example when $\sigma = 0.02$. Means and standard deviations of the estimates based on 50 independent runs if the estimates have i) equal weights and ii) weights proportional the final sample size. The results are obtained for different values of the asymptotic levels α, β and γ, see relations (26) and (27). The sample size was increased as $m_t \to m_t + m_t/3$, when $ALB \leq 0$.

α, β, γ	0.25	0.20	0.15	0.1–0.25–0.1	0.1
a_b	2.9728	2.9740	2.9745	2.9744	2.9749
	$(1.61 \cdot 10^{-3})$	$(1.69 \cdot 10^{-3})$	$(1.42 \cdot 10^{-3})$	$(1.43 \cdot 10^{-3})$	$(1.39 \cdot 10^{-3})$
a_p	2.9736	2.9747	2.9751	2.9750	2.9755
	$(1.45 \cdot 10^{-3})$	$(1.51 \cdot 10^{-3})$	$(1.31 \cdot 10^{-3})$	$(1.28 \cdot 10^{-3})$	$(1.23 \cdot 10^{-3})$
P_p	0.8153	0.8153	0.8153	0.8153	0.8153
	$(7.00 \cdot 10^{-6})$	$(6.00 \cdot 10^{-6})$	$(6.00 \cdot 10^{-6})$	$(5.00 \cdot 10^{-6})$	$(5.00 \cdot 10^{-6})$
μ^{-1}	100.2758	100.2700	100.2684	100.2671	100.2637
	(0.0094)	(0.0094)	(0.0084)	(0.0087)	(0.0088)
σ_b	0.0477	0.0477	0.0477	0.0477	0.0477
	$(4.80 \cdot 10^{-5})$	$(3.50 \cdot 10^{-5})$	$(2.50 \cdot 10^{-5})$	$(2.00 \cdot 10^{-5})$	$(2.50 \cdot 10^{-5})$
σ_p	0.0504	0.0504	0.0504	0.0504	0.0504
	$(3.70 \cdot 10^{-5})$	$(3.80 \cdot 10^{-5})$	$(3.10 \cdot 10^{-5})$	$(2.90 \cdot 10^{-5})$	$(2.21 \cdot 10^{-5})$
ρ	0.8283	0.8283	0.8283	0.8284	0.8284
	$(2.96 \cdot 10^{-3})$	$(3.00 \cdot 10^{-3})$	$(2.46 \cdot 10^{-3})$	$(2.25 \cdot 10^{-3})$	$(1.60 \cdot 10^{-3})$
	Weighted averaging				
a_b	2.9726	2.9737	2.9744	2.9742	2.9745
	$(1.44 \cdot 10^{-3})$	$(1.37 \cdot 10^{-3})$	$(1.01 \cdot 10^{-3})$	$(1.27 \cdot 10^{-3})$	$(1.14 \cdot 10^{-3})$
a_p	2.9734	2.9744	2.9750	2.9748	2.9751
	$(1.29 \cdot 10^{-3})$	$(1.22 \cdot 10^{-3})$	$(1.00 \cdot 10^{-3})$	$(1.15 \cdot 10^{-3})$	$(1.02 \cdot 10^{-3})$
P_p	0.8153	0.8153	0.8153	0.8153	0.8153
	$(6.00 \cdot 10^{-6})$	$(5.00 \cdot 10^{-6})$	$(6.00 \cdot 10^{-6})$	$(5.00 \cdot 10^{-6})$	$(5.00 \cdot 10^{-6})$
μ^{-1}	100.2767	100.2706	100.2674	100.2685	100.2660
	(0.0080)	(0.0085)	(0.0069)	(0.0078)	(0.0067)
σ_b	0.0477	0.0477	0.0477	0.0478	0.0477
	$(3.60 \cdot 10^{-5})$	$(2.80 \cdot 10^{-5})$	$(2.00 \cdot 10^{-5})$	$(1.70 \cdot 10^{-5})$	$(2.20 \cdot 10^{-5})$
σ_p	0.0504	0.0504	0.0504	0.0504	0.0504
	$(3.10 \cdot 10^{-5})$	$(3.10 \cdot 10^{-5})$	$(2.60 \cdot 10^{-5})$	$(2.30 \cdot 10^{-5})$	$(1.90 \cdot 10^{-5})$
ρ	0.8283	0.8282	0.8283	0.8284	0.8284
	$(2.47 \cdot 10^{-3})$	$(2.39 \cdot 10^{-3})$	$(2.12 \cdot 10^{-3})$	$(1.77 \cdot 10^{-3})$	$(1.54 \cdot 10^{-3})$

algorithm combined with the fact that the automated algorithm uses in an intelligent way the computational resources.

5 Application to a real dataset and model comparison

In this section, we present an application of our method with experimental data from the sugar-beet. The experimental protocol is detailed in [30]. This real-data case was presented in [46] to motivate the use of a hidden Markov model as the best choice among competing models. The current data contain mass measurements from 42 blades and petioles, assumed to have expansion durations $T = 10$. With this assumption all measurements correspond to leaves

Table 7: Descriptive statistics for the total sample size (TSS), the final sample size (FSS) and the number of iterations until convergence (Iter) corresponding to the tests given in Table 6.

α, β, γ	0.25	0.20	0.15	0.1–0.25–0.10	0.10
Min TSS	3685	6491	6359	9895	7193
Mean TSS	22413	29272	36910	41078	59441
Max TSS	89009	98637	166175	159998	235068
Min FSS	1066	1798	1964	2988	3861
Mean FSS	6314	7803	10780	12106	17465
Max FSS	27098	21869	39632	40842	66197
Min Iter	12	10	10	10	8
Max Iter	27	21	18	18	18

which have completed their expansion when the plant was cut. The measurements are given for reference in Table 8. The parameters are divided into two categories, those which were calibrated

Table 8: A dataset from the sugar-beet plant. Mass measurements from 42 blades (bl) and petioles (pe).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
bl	0.021	0.069	0.084	0.138	0.246	0.414	0.604	0.85	0.892	0.99	1.398	1.627	1.568	1.774
pe	0.01	0.014	0.023	0.045	0.079	0.29	0.475	0.529	0.537	0.649	0.857	0.988	1.059	1.216
	15	16	17	18	19	20	21	22	23	24	25	26	27	28
bl	1.728	1.625	1.349	1.297	1.212	1.184	1.097	1.028	0.943	0.856	0.744	0.615	0.555	0.476
pe	1.317	1.263	1.154	1.204	1.134	1.106	1.056	0.964	0.904	0.889	0.797	0.687	0.655	0.532
	29	30	31	32	33	34	35	36	37	38	39	40	41	42
bl	0.422	0.361	0.326	0.277	0.238	0.191	0.179	0.15	0.124	0.117	0.079	0.089	0.106	0.095
pe	0.52	0.471	0.392	0.365	0.296	0.241	0.242	0.186	0.167	0.126	0.091	0.094	0.094	0.083

directly in the field and the unknown parameter θ that has to be estimated. In Table 9 we give the values of the fixed parameters and the initial values that we used for the parameters that have to be estimated (determined in a preliminary searching stage).

In Table 10 we present the parameter estimation results that we obtained with the MCMC-ECM and the SISR-ECM algorithm by fitting the real data with the model $\mathcal{M}_1$. The details of the implementation are given in Section 3. The parameter σ^2 represents a standard level of uncertainty for the mean biophysical model given by (3). This value of $\sigma = 0.1$ corresponds to the model which best fits the data as shown in [46]. In Table 11 we present the parameter estimation results that we obtained with the automated MCMC-ECM algorithm. The details of the implementation are given in Section 4. Moreover, in Table 12 the corresponding descriptive statistics for the total sample size (TSS), the final sample size (FSS) and the number of ECM iterations (Iter) until convergence are given.

We remark that the mean parameter estimates that we obtained with the SISR-ECM, the MCMC-ECM and the automated MCMC-ECM algorithm are similar. We reach the same conclusion even if we use the averaging techniques. Nevertheless, notice in Table 10 that the standard deviations from the mean estimates among independent realizations are roughly from two to six times smaller with the MCMC-ECM than the SISR-ECM. Since the number of in-

Table 9: Initial values for both unknown and fixed parameters used to initialize the algorithms in the real data case, where σ_b, σ_p and ρ are the standard deviations and the correlation coefficient of the measurement error model (see Section 2 for the explanation of the other parameters)

param.	unknown	param.	known	param.	known
a_b	2.829	σ	0.1	e_b	0.0083
a_p	1.813	a_r	3.1	s^{pr}	500
p_p	0.8139	p_r	329.48	b_b	2
μ^{-1}	97.95	k_B	0.7	b_p	2
σ_b	0.076	t_r	60	b_r	2
σ_p	0.059	T	10		
ρ	0.136	q_0	0.003		

dependent realizations decreases linearly the variance, this means that we need at least four times less realizations with the MCMC to have approximately the same precision with the SISR for some of these parameters. This is an important advantage of the MCMC-ECM since the CPU time needed for a single run is approximately the same for both algorithms. The same conclusion holds for the automated MCMC-ECM algorithm as we can see in Table 11. The choice $\alpha = \beta = \gamma = 0.25$ results in significantly less standard deviations than the SISR, and slightly more than the non-automated MCMC. When the parameters α, β and γ decrease, then as expected the standard deviations decrease since the final and the total Monte Carlo sample size increases. Notice also in Table 12 that smaller values of α, β and γ decrease the total number of ECM steps until convergence. The advantages of the automated algorithm cannot be counterbalanced by using averaging in the non-automated algorithm as we can see in Table 10. Consequently, the choice of a single run of an automated MCMC-ECM is very reasonable even with the choice $\alpha = \beta = \gamma = 0.25$ and, depending on the desired accuracy, a small number of independent runs could be combined to obtain the weighted mean estimates. Furthermore, the automated algorithm makes indeed an intelligent use of Monte Carlo resources and there is no need to determine a priori the total number of ECM steps and how the Monte Carlo sample size should be increased.

In the last part of this section we present the results of the model comparison when fitting the experimental data presented in Table 8. Two types of models, referred to as models $\mathcal{M}_1$ and $\mathcal{M}_2$, were considered in this paper and their hidden Markov formulation is given by Proposition 1. For each model, we distinguished the cases when the correlation coefficient ρ between the mass measurement errors of the blade and the petiole is a free parameter that has to be estimated (model $\mathcal{M}_1$ and $\mathcal{M}_2$) or is null (model $\mathcal{M}_1^*$ and $\mathcal{M}_2^*$). In the latter cases we have one parameter less to estimate. We run the automated MCMC-ECM for all these models with $\alpha = \beta = \gamma = 0.25$ and the obtained results are presented in Table 13. We also give the estimated corrected Akaike information criterion (AICc) and the Bayesian information criterion (BIC) for all the models that we tested (see, e.g., [1] and the references therein). We infer that the model $\mathcal{M}_1^*$ performed better and that the additive error in the mass measurements (models $\mathcal{M}_1^*$ and $\mathcal{M}_1$) is better adapted than the log-additive one (models $\mathcal{M}_2^*$ and $\mathcal{M}_2$) since both criteria best rank the model with the lowest value. Even though we have restricted ourselves to the comparison between these models, the comparison method is of course general, and could be applied to other formulations of the error models or of the functional models.

Table 10: Parameter estimation results based on the real dataset. Means and standard deviations of the estimates based on 50 independent runs with SISR and MCMC for three different averaging strategies.

	No averaging		Averaging 25		Averaging 50	
param.	SISR	MCMC	SISR	MCMC	SISR	MCMC
a_b	2.8319	2.8362	2.8285	2.8379	2.8291	2.8379
	(0.0211)	(0.0100)	(0.0261)	(0.0128)	(0.0256)	(0.0126)
a_p	1.8259	1.8287	1.8222	1.8297	1.8227	1.8297
	(0.0162)	(0.0072)	(0.0188)	(0.0091)	(0.0187)	(0.0090)
P_p	0.8147	0.8147	0.8147	0.8147	0.8147	0.8147
	$(3.97 \cdot 10^{-4})$	$(0.67 \cdot 10^{-4})$	$(1.42 \cdot 10^{-4})$	$(0.25 \cdot 10^{-4})$	$(1.49 \cdot 10^{-4})$	$(0.25 \cdot 10^{-4})$
μ^{-1}	98.0730	97.9902	98.1324	97.9714	98.1245	97.9721
	(0.3913)	(0.1877)	(0.4745)	(0.2336)	(0.4653)	(0.2327)
σ_b	0.0761	0.0761	0.0762	0.0761	0.0761	0.0761
	$(3.83 \cdot 10^{-4})$	$(2.75 \cdot 10^{-4})$	$(1.12 \cdot 10^{-4})$	$(0.56 \cdot 10^{-4})$	$(2.74 \cdot 10^{-4})$	$(1.08 \cdot 10^{-4})$
σ_p	0.05887	0.05889	0.0589	0.0589	0.0590	0.0590
	$(2.45 \cdot 10^{-4})$	$(0.67 \cdot 10^{-4})$	$(1.54 \cdot 10^{-4})$	$(0.52 \cdot 10^{-4})$	$(1.54 \cdot 10^{-4})$	$(0.50 \cdot 10^{-4})$
ρ	0.1260	0.1246	0.1268	0.1246	0.1266	0.1246
	$(6.87 \cdot 10^{-3})$	$(1.73 \cdot 10^{-3})$	$(2.78 \cdot 10^{-3})$	$(1.23 \cdot 10^{-3})$	$(2.98 \cdot 10^{-3})$	$(1.27 \cdot 10^{-3})$

6 Discussion

In this paper we proposed simulation techniques based on MCMC for parameter estimation via an ECM algorithm for a class of plant growth models which can be characterized by deterministic structural development and include process error in biomass production dynamics, initially introduced in [46]. The resulting estimation algorithm based on MCMC improves the one developed in [46], where the authors used SISR to perform the Monte Carlo E-step, by reducing significantly the variance of parameter estimates obtained by independent runs of the algorithm. Another important advantage of this algorithm as compared to the one proposed in [46] is that the organ masses of the last immature members can all be taken into account even for large expansion durations and this could be very important for improving the quality of parameter estimation. Moreover, the adaptation of the data–driven automated algorithm of [4] to our algorithm was shown to be a good solution for an intelligent use of Monte Carlo resources. Simulation studies from a synthetic example and a real dataset from the sugar-beet plant were used to illustrate the performance of the proposed algorithm. Two different types of hidden Markov models were described and tested on a real dataset for their fitting quality.

The resulting algorithm is very promising and can be further exploited for decision aid in agricultural science. In this direction, further effort is needed for the adaptation of this algorithm to other crop plants with deterministic organogenesis and for model comparison and validation. Furthermore, despite the interest in individual plant growth modeling, the genetic variability of plants, even of the same variety, can be very important and, if we add locally varying climatic effects, then the development of two plants in the same field could be highly different. Consequently, a population-based model could be more appropriate to describe the population dynamics and the inter-individual variability ([10]). We are currently studying an extension to the population level by coupling with a nonlinear mixed effects model ([28]). Another interesting perspective is to broaden the applicability of the proposed statistical methodology in plants with stochastic organogenesis (e.g. trees) where the total number of organs of each class at each

Table 11: Parameter estimation results based on the real dataset with the automated MCMC-ECM algorithm. Means and standard deviations of the estimates based on 50 independent runs if the estimates have i) equal weights and ii) weights proportional to the final sample size. The results are obtained for different values of the asymptotic levels α, β and γ, see relations (26) and (27).

	z25	z20	z15	z10-25	z10
a_b	2.8363	2.8347	2.8346	2.8347	2.8335
	(0.0122)	(0.0089)	(0.0091)	(0.0085)	(0.0070)
a_p	1.8284	1.8270	1.8264	1.9269	1.8254
	(0.0087)	(0.0064)	(0.0065)	(0.0060)	(0.0047)
P_p	0.8147	0.8147	0.8147	0.8147	0.8147
	$(0.64 \cdot 10^{-4})$	$(0.58 \cdot 10^{-4})$	$(0.56 \cdot 10^{-4})$	$(0.52 \cdot 10^{-4})$	$(0.63 \cdot 10^{-4})$
μ^{-1}	97.9941	98.0231	98.0197	98.0261	98.0389
	(0.2118)	(0.1647)	(0.1665)	(0.1567)	(0.1315)
σ_b	0.0761	0.0761	0.0761	0.0761	0.0761
	$2.25 \cdot 10^{-4})$	$(1.80 \cdot 10^{-4})$	$(1.65 \cdot 10^{-4})$	$(1.38 \cdot 10^{-4})$	$(1.57 \cdot 10^{-4})$
σ_p	0.0589	0.0589	0.0589	0.0589	0.0589
	$(0.94 \cdot 10^{-4})$	$(0.87 \cdot 10^{-4})$	$(0.78 \cdot 10^{-4})$	$(0.53 \cdot 10^{-4})$	$(0.63 \cdot 10^{-4})$
ρ	0.1254	0.1261	0.1265	0.1259	0.1271
	$(2.51 \cdot 10^{-3})$	$(2.28 \cdot 10^{-3})$	$(2.31 \cdot 10^{-3})$	$(2.12 \cdot 10^{-3})$	$(2.15 \cdot 10^{-3})$
	Weighted averaging				
a_b	2.8372	2.8349	2.8354	2.8345	2.8346
	(0.0120)	(0.0086)	(0.0104)	(0.0078)	(0.0068)
a_p	1.8290	1.8272	1.8272	1.8268	1.8264
	(0.0087)	(0.0062)	(0.0073)	(0.0054)	(0.0045)
P_p	0.8147	0.8147	0.8147	0.8147	0.8147
	$(0.57 \cdot 10^{-4})$	$(0.55 \cdot 10^{-4})$	$(0.51 \cdot 10^{-4})$	$(0.49 \cdot 10^{-4})$	$(0.47 \cdot 10^{-4})$
μ^{-1}	97.9824	98.0244	98.0077	98.0311	98.0220
	(0.2076)	(0.1590)	(0.1901)	(0.1473)	(0.1307)
σ_b	0.0761	0.0761	0.0761	0.0761	0.0761
	$(2.03 \cdot 10^{-4})$	$(1.54 \cdot 10^{-4})$	$(1.47 \cdot 10^{-4})$	$(1.18 \cdot 10^{-4})$	$(1.43 \cdot 10^{-4})$
σ_p	0.0589	0.0589	0.0589	0.0589	0.0588
	$(0.87 \cdot 10^{-4})$	$(0.73 \cdot 10^{-4})$	$(0.67 \cdot 10^{-4})$	$(0.49 \cdot 10^{-4})$	$(0.63 \cdot 10^{-4})$
ρ	0.1253	0.1260	0.1260	0.1257	0.1267
	$(2.62 \cdot 10^{-3})$	$(2.11 \cdot 10^{-3})$	$(2.09 \cdot 10^{-3})$	$(1.74 \cdot 10^{-3})$	$(1.86 \cdot 10^{-3})$

growth cycle is a random variable (see, e.g., [33]).

References

[1] T. Bengtsson and J.E. Cavanaugh. An improved Akaike information criterion for state-space model selection. *Computational Statistics & Data Analysis*, 50(10):2635–2654, 2006.

[2] J.G. Booth and J.P. Hobert. Maximizing Generalized Linear Mixed Model likelihoods with an Automated Monte Carlo EM algorithm. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 61(1):265–285, 1999.

[3] P. Bratley, B.L. Fox, and L.E. Schrage. *A Guide to Simulation*. Springer-Verlag, New York, 1987.

Table 12: Descriptive statistics for the total sample size (TSS), the final sample size (FSS) and the number of iterations until convergence (Iter) corresponding to the tests given in Table 11.

	z25	z20	z15	z10-25	z10
Min TSS	7376	3275	7452	4504	4530
Mean TSS	30173	30326	32949	41645	36744
Max TSS	66311	73335	122000	105438	230250
Min FSS	2675	1801	4071	3158	3561
Mean FSS	9570	12959	16816	19354	22487
Max FSS	20820	39527	51945	47450	160257
Min Iter	6	5	5	4	3
Max Iter	21	12	13	15	8

Table 13: MLE obtained with the models $\mathcal{M}_1$ (ρ as a free parameter), $\mathcal{M}_1^*$ (ρ=0), $\mathcal{M}_2$ (ρ as a free parameter) and $\mathcal{M}_2^*$ (ρ=0) for the sugar-beet dataset. In the last two columns the corrected Akaike information criterion (AICc) and the Bayesian information criterion (BIC) are estimated based on 100 samples of 5×10^5 independent evaluations. The standard deviations are given in parenthesis. The above criteria are given by $AICc = -2(\log \hat{L} - d) + 2d(d+1)/(n-d+1)$ and $BIC = -2\log \hat{L} + d \log n$, where d is the number of free parameters and n the sample size.

model	a_b	a_p	P_p	μ^{-1}	σ_b	σ_p	ρ	$\hat{AICc}$	$\hat{BIC}$
$\mathcal{M}_1^*$	2.836	1.852	0.8142	98.48	0.0750	0.0591	0.0000	-344.13 (0.03)	-330.63 (0.03)
$\mathcal{M}_1$	2.837	1.829	0.8147	97.98	0.0761	0.0589	0.1253	-342.17 (0.02)	-326.63 (0.02)
$\mathcal{M}_2^*$	3.019	2.044	0.8031	98.76	0.1585	0.2119	0.0000	-334.72 (0.03)	-321.23 (0.03)
$\mathcal{M}_2$	3.139	2.172	0.8051	96.83	0.1647	0.2114	-0.3380	-336.18 (0.04)	-320.64 (0.04)

[4] B. S. Caffo, W. Jank, and G. L. Jones. Ascent–based Monte Carlo Expectation–Maximization. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67:235–251, 2005.

[5] O. Cappé, E. Moulines, and T. Rydén. *Inference in Hidden Markov Models*. Springer, New York, 2005.

[6] G. Celeux and J. Diebolt. The SEM algorithm: A probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quarterly*, 2:73–82, 1985.

[7] J.S.K. Chan and A.Y.C. Kuk. Maximum likelihood estimation for probitlinear mixed models with correlated random effects. *Biometrics*, 53:86–97, 1998.

[8] P.-H. Cournède, V. Letort, A. Mathieu, M.-Z. Kang, S. Lemaire, S. Trevezas, F. Houllier, and P. de Reffye. Some parameter estimation issues in functional-structural plant modelling. *Mathematical Modelling of Natural Phenomena*, 6(2):133–159, 2011.

[9] P. de Reffye and B.G. Hu. Relevant choices in botany and mathematics for building efficient dynamic plant growth models: the greenlab case. In B.G. Hu and M. Jaeger, editors, *Plant Growth Models and Applications*, pages 87–107. Tsinghua University Press and Springer, 2003.

[10] P. de Reffye, S. Lemaire, N. Srivastava, F. Maupas, and P.-H. Cournède. Modeling inter-individual variability in sugar beet populations. In B.G. Li, M. Jaeger, and Y. Guo, editors, *3rd international symposium on Plant Growth and Applications(PMA09), Beijing, China*. IEEE, November 9-12 2009.

[11] B. Delyon, V. Lavielle, and E. Moulines. Convergence of a stochastic approximation version of the EM algorithm. *Annals of Statistics*, 27:94–128, 1999.

[12] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum Likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 39:1–38, 1977.

[13] A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Springer-Verlag, New-York, 2001.

[14] J. M. Flegal and G.L. Jones. Batch means and spectral variance estimators in Markov chain Monte Carlo. *The Annals of Statistics*, 2(38):1034–1070, 2010.

[15] G. Fort and E. Moulines. Convergence of the Monte Carlo expectation maximization for curved exponential families. *The Annals of Statistics*, (31):1220–1259, 2003.

[16] A.E. Gelfand and A.F.M. Smith. Sampling based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409, 1990.

[17] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[18] C. J. Geyer. Practical Markov Chain Monte Carlo. *Statistical Science*, 7(4):473–483, 1992.

[19] N. Gordon, D. Salmond, and A. F. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F, Radar Signal Process.*, 140(2):107–113, 1993.

[20] Y. Guo, Y.T. Ma, Z.G. Zhan, B.G. Li, M. Dingkuhn, D. Luquet, and P. de Reffye. Parameter optimization and field validation of the functional-structural model GREENLAB for maize. *Annals of Botany*, 97:217–230, 2006.

[21] W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[22] W. Jank. Quasi-Monte Carlo sampling to improve the efficiency of Monte Carlo EM. *Computational Statistics & Data Analysis*, 48(4):685 – 701, 2005.

[23] W. Jank. Stochastic Variants of EM: Monte Carlo, Quasi–Monte Carlo and More. In *Proceedings of the American Statistical Association*, 2005.

[24] W. Jank. The EM algorithm, Its Stochastic Implementation and Global Optimization: Some Challenges and Opportunities for OR. In F. Alt, M. Fu, and B. Golden, editors, *Topics in Modeling, Optimization and Decision Technologies: Honoring Saul Gass' Contributions to Operation Research*, pages 367–392. Springer-Verlag, 2006.

[25] G.L. Jones, M. Haran, Caffo B.S., and Neath. R. Fixed-width output analysis for Markov chain Monte Carlo. *Journal of the American Statistical Association*, 101.476:1537–1547, 2006.

[26] G.L. Jones and J.P. Hobert. Honest exploration of intractable probability distributions via Markov chain Monte Carlo. *Statistical Science*, 16(4):312–334, 2001.

[27] A. Jullien, A. Mathieu, J.-M. Allirand, A. Pinet, P. de Reffye, P.-H. Cournède, and B. Ney. Characterisation of the interactions between architecture and source:sink relationships in winter oilseed rape (brassica napus l.) using the greenlab model. *Annals of Botany*, 107(5):765–779, 2011.

[28] E. Kuhn and M Lavielle. Maximum likelihood estimation in nonlinear mixed effects models. *Computational Statistics and Data Analysis*, 49:1020–1038, 2005.

[29] K. Lange. A Gradient Algorithm Locally Equivalent to the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(2):425–437, 1995.

[30] S. Lemaire, F. Maupas, P.-H. Cournède, and P. de Reffye. A morphogenetic crop model for sugar-beet (*beta vulgaris l.*). In *International Symposium on Crop Modeling and Decision Support: ISCMDS 2008, April 19-22, 2008, Nanjing, China*, 2008.

[31] R.A. Levine and G. Casella. Implementations of the Monte Carlo EM algorithm. *Journal of Computational and Graphical Statistics*, 10(3):422–439, 2001.

[32] R.A. Levine and J. Fan. An Automated (Markov chain) Monte Carlo EM algorithm. *Journal of Statistical Computation and Simulation*, 74(5):349–360, 2004.

[33] C. Loi and P.-H. Cournède. Generating functions of stochastic L-systems and application to models of plant development. *Discrete Mathematics and Theoretical Computer Science Proceedings*, AI:325–338, 2008.

[34] A. Mathieu, P.-H. Cournède, V. Letort, D. Barthélémy, and P. de Reffye. A dynamic model of plant growth with interactions between development and functional mechanisms to study plant structural plasticity related to trophic competition. *Annals of Botany*, 103(8):1173–1186, 2009.

[35] C.E. McCulloch. Maximum likelihood variance components estimation for binary data. *Journal of the American Statistical Association*, 89:330–335, 1994.

[36] C.E. McCulloch. Maximum likelihood algorithms for generalized linear mixed models. *Journal of the American Statistical Association*, 92:162–170, 1997.

[37] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons Inc., 2008.

[38] X.-L. Meng and D. B. Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80:267–278, 1993.

[39] Rosenbluth A. W. Rosenbluth M. N. Teller A. H. Metropolis, N. and Teller E. Equation of state calculation by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[40] S.P. Meyn and R.L. Tweedie. *Markov Chains and Stochastic Stability*. Springer–Verlag, 1993.

[41] P. Mykland, L. Tierney, and B. Yu. Regeneration in Markov Chain Samplers. *Journal of the American Statistical Association*, 90:233–241, 1995.

[42] M. B. Priestley. *Spectral Analysis and Time Series*. Academic, London, 1981.

[43] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[44] C. P. Robert, T. Ryden, and D.M. Titterington. Convergence controls for MCMC algorithms, with applications to Hidden Markov Chains. *Journal of Statistical Computation and Simulation*, 64:327355, 1999.

[45] R. Sievänen, E. Nikinmaa, P. Nygren, H. Ozier-Lafontaine, J. Perttunen, and H. Hakula. Components of a functional-structural tree model. *Annals of Forest Sciences*, 57:399–412, 2000.

[46] S. Trevezas and P.-H. Cournède. A sequential Monte Carlo approach for MLE in a plant growth model. *Journal of Agricultural, Biological, and Environmental Statistics*, accepted.

[47] J. Vos, L.F.M. Marcelis, and J.B. Evers. Functional-structural plant modelling in crop production. In J. Vos, L.F.M. Marcelis, P.H.B. de Visser, P.C. Struik, and J.B. Evers, editors, *Functional-structural plant modelling in crop production, Wageningen*, volume Chapter 1. Springer, 2007.

[48] G. Wei and M. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85:699–704, 1990.

Appendix

Proof of Proposition 2.
In order to simplify the proof we will change the state variables of the model $\mathcal{M}_2$. By setting $R_n = \log Q_n$ and $Z_n = \log Y_n$ we can rewrite (10) and (11) as follows:

$$\begin{aligned} R_{n+1} &= \log F_n(R_{(n-T+1)^+:n}; \mu, p_{al}) + W_n, & (30)\\ Z_n &= \log G_n(R_{n:(n+T-1)}; p_{al}) + V_n. & (31) \end{aligned}$$

Now, let us analyze the Q-function of the model. Let us also write F_n given by (3) as $F_n = \mu K_n$. In the rest, we identify the functions K_n and G_n (see (6)) with the induced random variable $K_n(\theta_2)$ and the induced random vector $G_n(\theta_2)$ respectively, for an arbitrary $\theta_2 \in \Theta_2$, where Θ_2 is an appropriate euclidean subset. By the assumptions of the model $\mathcal{M}_2$ and equations (30) and (31) we have:

$$\begin{aligned} \mathcal{Q}(\theta;\theta') &= \mathbb{E}_{\theta'}\left[\log p_\theta(R_{0:N}, z_{0:N}) \mid z_{0:N}\right] \\ &= \sum_{n=1}^{N} \mathbb{E}_{\theta'}\left[\log p_\theta(R_n | R_{(n-T)^+:n-1}) \mid z_{0:N}\right] \\ &\quad + \sum_{n=0}^{N} \mathbb{E}_{\theta'}\left[\log p_\theta(z_n | R_{n:(n+T-1)\wedge N}) \mid z_{0:N}\right] \\ &= C(\theta_2;\theta') + \mathcal{Q}_1(\mu, \sigma^2, \theta_2; \theta') + \mathcal{Q}_2(\Sigma, \theta_2; \theta'), & (32) \end{aligned}$$

where

$$\mathcal{Q}_1(\mu, \sigma^2, \theta_2; \theta') = -\frac{N}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{n=1}^{N} \mathbb{E}_{\theta'}\left[(R_n - \log K_{n-1}(\theta_2) - \log\mu)^2 \mid z_{0:N}\right],$$

$$\begin{aligned}\mathcal{Q}_2(\Sigma,\theta_2;\theta') &= -\frac{N+1}{2}\log(\det\Sigma)\\ &\quad -\frac{1}{2}\sum_{n=0}^{N}\mathbb{E}_{\theta'}\left[\big(z_n-\log G_n(\theta_2)\big)^\top\Sigma^{-1}\big(z_n-\log G_n(\theta_2)\big)\mid z_{0:N}\right],\end{aligned}$$

and $C(\theta_2;\theta')$ is independent of θ_1.

Note that for fixed θ_2 the initial maximization problem of $\mathcal{Q}$ w.r.t. θ_1 can be separated into two distinct maximization problems of $\mathcal{Q}_1$ and $\mathcal{Q}_2$ w.r.t. (μ,σ^2) and Σ respectively. By maximizing $\mathcal{Q}_1$ we get easily (18) and (20) and by maximizing $\mathcal{Q}_2$ we get (19). In the latter case the proof is the same as in the case of an additive measurement error model (with the transformed variables) and a detailed proof can be found in [46], Web Appendix C.